Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam
(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: 24-07-2026

Course Code	**UCS3511**	Course Name	**Networks Lab**						
Course Type	**Lab**	Course Category	**Professional Core (PC)**	**L**	**T**	**P**	**TW & SL**	**TH**	**C**
				0	**0**	**45**	**0**	**45**	**1.5**
Regulation	**2024**		**Academic Year & Batch**	**2026-27 & 2024-28**					
Degree and Branch			**B.E. Computer Science & Engineering**						
Semester			**V**	**Section**	**A, B & C**	**K3, CO2**	**PI:** 1.1.2, 1.3.1, 1.4.1, 2.1.2, 2.1.3, 2.3.1, 3.1.1, 4.3.1, 5.1.1, 12.2.1, 12.3.1, 12.4.1 & 13.4.1		
Assignment No.	**11**	**Assigned Date**	**28-09-2026**	**Due Date**	**06-10-2026**				

Lab Assignment: Simulation of CSMA/CA Protocol Using SimPy

Topic: Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
Application: Wireless LAN Communication

Scenario

A wireless LAN consists of multiple stations sharing a common wireless communication channel. Each station must sense the channel before transmission and use a random backoff mechanism when the channel is busy or when multiple stations attempt transmission simultaneously. Develop a **SimPy-based simulation of the CSMA/CA protocol** to implement carrier sensing, random backoff, transmission, collision detection, and retransmission.

Tasks

1a. Create Wireless Nodes

Develop a SimPy model with **5 wireless nodes**. Each node should:

- Generate data packets at specified intervals.
- Sense the shared wireless channel before transmission.
- Select a random backoff value when the channel is busy.
- Wait for the backoff period before attempting transmission.
- Transmit the packet when the channel becomes available.

1b. Implement Shared Wireless Channel

Create a shared wireless channel using SimPy. The channel should:

- Maintain **BUSY/IDLE** states.
- Allow a node to transmit when the channel is idle.
- Prevent transmission when the channel is busy.
- Record successful transmissions and collisions.

1c. Implement CSMA/CA Backoff

Implement the CSMA/CA backoff mechanism using:

- Minimum Contention Window (CWmin) = 4
- Maximum Contention Window (CWmax) = 16
- Slot time = 1 ms
- Random backoff selection within the contention window.
- Backoff freezing when the channel becomes busy.
- Resumption of backoff when the channel becomes idle.

1d. Implement Collision and Retransmission

When two or more nodes attempt transmission simultaneously:

- Identify the transmission as a collision.
- Increase the contention window.
- Retransmit the packet after selecting a new backoff value.
- Reset the contention window after a successful transmission.
- Limit the maximum number of retransmission attempts to **4**.

1e. Simulation Parameters

Use the following parameters:

Parameter	Value
Number of Nodes	5
Packet Size	1000 Bytes
CWmin	4
CWmax	16
Slot Time	1 ms
Transmission Time	10 ms
Maximum Retransmissions	4
Simulation Time	10 seconds
Packet Generation Rate	20 packets/sec/node

1f. Test Cases

Implement and execute the following test cases:

Test Case	No. of Nodes	CW Range	Traffic Load	Expected Observation
TC1	2	4–16	Low	Successful transmissions with few collisions
TC2	5	4–16	Low	Multiple transmissions and occasional collisions
TC3	5	2–8	Medium	Increased collision and retransmission
TC4	5	8–16	Medium	Longer backoff before transmission
TC5	10	4–16	High	Increased contention and retransmissions

Expected Output

For each node, display:

- Packet generation time
- Channel sensing status
- Selected backoff value
- Transmission start time
- Transmission completion time
- Collision status
- Retransmission count
- Successful transmission status

Example:

```
Node N1: Channel IDLE
Node N1: Selected Backoff = 3 slots
Node N2: Channel IDLE
Node N2: Selected Backoff = 3 slots

Node N1: Transmission Started
Node N2: Collision Detected
Node N2: Retransmission Attempt = 1
Node N2: New CW = 8
Node N1: Transmission Successful
```

Sample Input

```
Number of Nodes          : 5
Packet Size              : 1000 Bytes
CWmin                    : 4
CWmax                    : 16
Slot Time                : 1 ms
Transmission Time        : 10 ms
Maximum Retransmissions  : 4
Simulation Time          : 10 seconds
Packet Generation Rate   : 20 packets/sec/node
```

Sample Output

```
CSMA/CA Simulation Started

Node N1: Packet generated at 0.120 s
Node N1: Channel IDLE
Node N1: Backoff = 2 slots
Node N1: Transmission started at 0.122 s
Node N1: Transmission successful at 0.132 s

Node N2: Packet generated at 0.185 s
Node N2: Channel BUSY
Node N2: Backoff = 4 slots
Node N2: Waiting for channel...

Node N3: Packet generated at 0.241 s
Node N3: Channel IDLE
Node N3: Backoff = 1 slot
Node N3: Transmission started
Node N3: Transmission successful

Node N4: Packet generated at 0.315 s
Node N5: Packet generated at 0.316 s
Node N4: Backoff = 2 slots
Node N5: Backoff = 2 slots
Node N4: Transmission started
Node N5: Collision detected

Node N5: Retransmission Attempt = 1
Node N5: CW increased from 4 to 8
Node N5: New Backoff = 5 slots
Node N5: Transmission successful

CSMA/CA Simulation Completed
Total Packets Generated = 100
Successful Transmissions = 94
Total Collisions = 18
Total Retransmissions = 21
```

Performance Observation Table

Test Case	No. of Nodes	CW Range	Packets Generated	Successful Packets	Collisions	Retransmissions	Avg. Backoff (ms)
TC1	2	4–16	40	39	3	3	2.8
TC2	5	4–16	100	94	18	21	5.6
TC3	5	2–8	100	87	32	38	3.1
TC4	5	8–16	100	96	12	14	8.4
TC5	10	4–16	200	171	67	79	7.3

Note: These are **sample/reference values**, not fixed expected outputs. Since CSMA/CA uses random backoff, students' actual values can differ when the SimPy program is executed. For reproducible lab results, a fixed random seed may be specified.

Sample Node-wise Output Summary

Node	Packets Generated	Successful	Collisions	Retransmissions
N1	20	19	3	3
N2	20	19	4	4
N3	20	18	5	6
N4	20	19	3	4
N5	20	19	3	4
Total	**100**	**94**	**18**	**21**

Deliverables

1. SimPy program implementing CSMA/CA.
2. Separate functions/processes for **node, channel, carrier sensing, backoff, transmission, and retransmission**.
3. Execution results for all five test cases.
4. Completed performance observation table.
5. Brief conclusion based on the simulation results.

Learning Outcome

By completing this experiment, students will be able to **apply the CSMA/CA protocol using SimPy to model carrier sensing, random backoff, collision handling, and retransmission in a wireless LAN environment.**